Preface

This book imparts a working knowledge of procedural approaches in texturing, modeling, shading, and animation. These include two-dimensional and solid texturing, hypertextures, volume density functions, and fractals. Readers are provided with the details often omitted from technical papers, enabling them to explore how the procedures are designed to produce realistic imagery. The reader will gain not only a powerful toolbox of procedures upon which to build a library of procedural textures and objects, but also a better understanding of how these functions work and how to design them. With procedures like *noise* and *turbulence* and an understanding of procedural design techniques, the reader will be able to design more complex procedures to produce realistic textures, gases, hypertextures, landscapes, and planets. The procedural techniques are explained not by people who have read some technical papers and tried to decipher them, but by the people who developed them, wrote the seminal papers in the area, and have worked with procedural design for many years.

Procedural Techniques in Computer Graphics

Procedural modeling, texturing, and shading is an area of active research and growing importance in computer graphics and animation. Procedural techniques have been used for many years to produce textures for objects. With the introduction of three-dimensional texturing techniques (solid texturing) by Ken Perlin, Darwyn Peachey, and Geoffrey Gardner in 1985, the use of procedural techniques exploded. Realistic images containing marble, wood, stone, and clouds were now possible. Procedural techniques became an area of active research in computer graphics. Many programmers and researchers developed their own procedures to simulate natural materials and natural phenomena. What was lacking, however, was a clear understanding of the design of procedural techniques and of the primitive stochastic functions that were used to produce these amazing images. Since the mid 1980s, the use of procedural techniques has grown rapidly, and they can now be used to actually define the geometry of objects such as water, fire, gases, planets, and tribbles.

The use of procedural techniques is not limited to still images; they have been used successfully for animation and recently for the simulation and animation of natural phenomena such as fog, fire, water, and atmospheric patterns. The animation of procedural models requires knowledge of both animation principles and the characteristics of the procedural model. Procedural animation is a powerful technique for producing complex, realistic motion.

Our Objective

The objective of this book is to provide the reader with an understanding and working knowledge of several uses of procedural techniques in texturing, modeling, and animation. This book describes the current state of procedural techniques and provides the reader with the challenge and information necessary to extend the state of the art. The reader will gain the following from the book:

- A thorough understanding of procedural techniques for solid texturing.
- An insight into different design approaches used by the authors in designing procedures.
- A toolbox of procedures and basic primitive functions (noise, turbulence, etc.) to produce realistic images.
- An understanding of several advanced procedural approaches for modeling object geometry (hypertextures, gases, fractals).
- An introduction to animating these procedural objects and textures.

Development of This Book

This book is an expansion of the courses on *Procedural Modeling and Rendering Techniques* that we have taught at SIGGRAPH '92 and SIGGRAPH '93. At SIGGRAPH '91, Darwyn Peachey and David Ebert first discussed the need for a course to explain how texture and modeling procedures are designed to create impressive images of wood and marble objects, gases, landscapes, and planets. There were some classic papers on these topics, but they were lacking in several respects. First of all, not enough detail was given to enable the reader to reproduce the results in most of the papers. Second, if an image could be reproduced, the reader still didn't know how to modify the procedure to get a different effect. Finally, the reason why a procedure produced a given image was unclear. There seemed to be some "magic" behind the development of these procedures. From this discussion, our course at SIGGRAPH '92 arose. There was great demand for the course at both SIGGRAPH '92 and SIGGRAPH '93. We have received useful feedback, thanks, and requests for more material on this topic from the attendees of these courses.

With the success of these courses, we decided to produce this book. This book is similar in nature to our course notes, but greatly expanded and revised. It describes the procedural approaches of five researchers in the field of computer graphics. The authors slightly overlap their discussion to provide different viewpoints on important concepts such as the fractal noise function.

Source Code

All of the source code for the examples in this book are contained in the diskette supplied with this book. The source code is also available for anonymous ftp on the Internet. The ftp site name is "`archive.cs.umbc.edu`"and the directory name is "`texture`".

Acknowledgments

First, we wish to thank Eric Haines and Rick Parent for reviewing drafts of this book. We also wish to thank the attendees of our SIGGRAPH courses for feedback and suggestions that led to the development of this book. Julie Ebert, Judy Peachey, Rick Sayre, and Susan Wrights have been of immeasurable help in reviewing parts of this book. Holly Rushmeier has helped with figure creation, Larry Gritz has provided Renderman translations for Ken Musgrave's code, Tom Deering has helped Darwyn Peachey with his Illustrator figures, Richard Chang has helped with file transfers, and Benjamin Zhu, Xue Dong Wong, Rajesh Raichoudhury, and Ron Espiritu have helped Ken Perlin in the preparation of his chapter. We also wish to thank our families, friends, and co-workers for support during the development of this book. Finally, we wish to thank Jenifer Niles, Lance Coderre, Karen Pratt, Cindy Kogut, and the staff at AP PROFESSIONAL for help in producing this final volume.

David S. Ebert

Author Addresses

Our current physical and e-mail addresses are given below.

Dr. David S. Ebert
Computer Science Department
University of Maryland Baltimore County
5401 Wilkens Ave.
Baltimore, MD 21228-5398
ebert@cs.umbc.edu

Dr. F. Kenton Musgrave
Computer Science Department
The George Washington University
20101 Academic Way
Ashburn, VA 22011
musgrave@seas.gwu.edu

Darwyn Peachey
Pixar
1001 West Cutting
Richmond, CA 94804
peachey@pixar.com

Dr. Ken Perlin
NYU Media Research Lab
715 Broadway, Rm 1224
NY, NY 10003
perlin@nyu.edu

Steven Worley
405 El Camino Real Suite 121
Menlo Park, CA 94025
spworley@netcom.com

1 Introduction

David S. Ebert and F. Kenton Musgrave

Why Proceduralism

Procedural techniques have been used for many years in computer graphics to produce realistic textures (marble, wood, stone, wallpaper, bricks) and objects (water, smoke, steam, fire, planets, tribbles). Obviously, procedural techniques have many uses in computer graphics and animation. They are a cost-effective alternative to physical simulation. In simulating natural phenomena, it is impossible or impractical in most cases to develop physically accurate models. Procedural models are a tractable alternative.

Procedural models also offer flexibility. The designer of the procedures can capture the essence of the object, phenomenon, or motion without being constrained by the complex laws of physics. Procedural techniques allow the inclusion of any amount of physical accuracy into the model that is desired. The designer may produce a wide range of effects, from accurately simulating natural laws to purely artistic effects.

From the above discussion, it is clear that proceduralism is a powerful paradigm for image synthesis. In a procedural approach, rather than explicitly specifying and storing all the complex details of a scene or sequence, we *abstract* them into a function or an algorithm (i.e., a *procedure*) and evaluate that procedure when and where needed. We gain a savings in storage, as the details are no longer explicitly specified but rather implicit in the procedure, and shift the time requirements for specification of details from the programmer to the computer. We also gain the power of parametric control, allowing us to assign to a parameter a meaningful concept (e.g., a number that makes mountains "rougher" or "smoother"). Finally, we gain the serendipity inherent in procedural techniques: we are often pleasantly surprised by the unexpected behaviors of procedures, particularly stochastic procedures.

Some aspects of image synthesis are by nature procedural, that is, they can't practically be evaluated in advance: view-dependent specular shading and atmospheric effects are examples. This makes proceduralism in computer graphics hard to qualify. What we are concerned with in this book is primarily *procedural textures* and their extensions to procedural volume objects. Launched by Gardner, Peachey, and Perlin at SIGGRAPH in 1985, the use of procedural textures has since been expanded and embellished by Ebert, Lewis, Musgrave, Saupe, Sims, and Van Wijk.

Aim of This Book

This book gives the reader a working knowledge of several procedural texturing, modeling, and animation techniques, including two-dimensional texturing, solid texturing, hypertextures, volume density functions, and fractal algorithms. We provide the reader with the details of these techniques, which are often omitted from technical papers, including useful and practical guidelines for selecting parameter values.

This book provides a toolbox of specific procedures and basic primitive functions (noise, turbulence, etc.) to produce realistic images. An in-depth description of noise functions is presented, accompanied by several implementations, and a spectral comparison of the different noise functions. Some of the procedures presented can be used to create realistic images of marble, brick, fire, steam, smoke, stone, and planets.

Organization

This book follows a progression in the use of procedural techniques from procedural texturing to volumetric procedural objects, and finally to fractals. In each chapter, the concepts are illustrated with a large number of example procedures. These procedures are presented in C code segments or in the RenderMan shading language.

The details of the design of these procedures are also explained to aid the reader in gaining insights into the different procedural design approaches taken by the authors. The reader should therefore be able to reproduce the images presented in this book and extend the procedures presented to produce a myriad of effects.

Darwyn Peachey describes how to build procedural textures in Chapter 2. This discussion includes two-dimensional texturing and solid texturing. Two important procedures that are used throughout this book are described in this chapter: *noise* and *turbulence*. Aliasing problems of procedural techniques are described and several antialiasing techniques suitable for procedural approaches are presented and compared to aid the reader in producing realistic, alias-free images.

The design of procedural textures is also the subject of Chapter 3 by Steve Worley. This chapter contains useful guidelines and tricks to produce interesting textures efficiently. Steve also describes some additional approaches to antialiasing.

The next two chapters by David Ebert describe how turbulence and solid texturing can be extended to produce images and animations of three-dimensional gases (particulate volumes). Chapter 4 concentrates on techniques to produce still images of volumetric "gases"(steam, fog, smoke) and Chapter 5 discusses how to animate these three-dimensional volumes and also how to animate solid textures and hypertextures.

Chapter 6 by Ken Perlin discusses other types of volumetric procedural objects: hypertextures and surflets. Ken also gives a detailed explanation of his famous *noise* function and its implementation, as well as rendering of procedural objects and antialiasing. Finally, this chapter discusses the use of high-level, nondeterministic "texture" to create gestural motion for synthetic actors. By blending textural controls, the apparent emotional state of a synthetic actor can be created.

Finally, Chapters 7, 8, and 9 by Ken Musgrave describe fractals and their use in creating realistic landscapes and planets. Ken begins by giving a brief introduction to fractals and then discusses fractal textures, fractal landscapes, and finally, fractal planets.